

R E M A R K S

In his Interview Summary the Examiner recognizes "two areas identified to have potentials for distinguishing over the applied art: the capabilities to read requested data from a specific location on the desired information source (documents such as a spreadsheet, word processing, and web page); and the ordinary expression is being read/written alongside the fetched requested data on the same window."

P R E S E N T D I S C L O S U R E

The presently disclosed software object can simultaneous display side-by-side entries that originated from very different sources, namely: (a) user entries alongside (b) automated entries that are (i) specified by those user entries; and (ii) fetched from a database or the like.

Specifically, the present specification describes selecting or pointing to a plurality of ordinary cells in a spreadsheet that the user typed into a spreadsheet.¹ The specification also describes how a user selects a database source or some other data source accessible over, for example, the Internet.² The specification further describes how a user can specify the data types desired from such sources.³

Using the example of Figure 4, if nothing further is done the only data that will

¹ Page 19, line 17 through page 20, line 1; and page 22, line 31 through page 23, line 4.

² Page 20, lines 16-24.

³ Page 20, lines 26-30 .

be displayed is the list of data appearing in the column containing manual entry FI. But the automated data will be fetched and displayed when the user clicks button 30 of Figure 5 (for an update, the user clicks button 80 of Figure 7).⁴

Now, the disclosed software object reads the plurality of symbols in the column containing entry FI.⁵ The software object then uses the symbols just read as a definition for the data to be fetched from an outside source. The fetched data is used to fill in the other blank columns.⁶

Accordingly, after setting up the display using the dialog box of Figure 5, the underlying application's display screen can then function almost like an ordinary display screen in the application. With the example of Table 1 of page 23 of the specification, the user can edit the first column, changing IBM to GM, and changing GE to STX, for example. After clicking button 80 the data on these two rows will change automatically. However, the user will have a sense of having stayed in the underlying application (i.e., having started in Excel and remained in Excel).

Note the software object can operate in environments other than a spreadsheet.

⁴ Page 21, lines 11-14; and page 26, lines 2-5.

⁵ Page 21, lines 14-16.

⁶ Page 2, lines 22-27.

MOSELEY

This reference describes the process for designing forms in an Access database. Basically, database forms are set up in a design mode, which forms are then populated with data in the subsequent ordinary operating mode. The system can operate in the design mode using either a form "wizard" (pp. 874-877) or the "form design screen" on page 879. The form design screen allows the designer to open a field list window and drag fields into position on the form being designed. The form design screen also has a toolbar with control objects. The designer can place a control object on the evolving form by clicking on a control object on the toolbar and then clicking at a destination on the form.

This form design screen can set up writing operations, but cannot set up reading operations of the type disclosed in the present application. Specifically, the form design screen allows the designer to specify where information from the database will be written, but the design screen cannot be used to specify which ordinary expression occurring during the subsequent ordinary operating mode will be read as a request for information that will be fetched and displayed alongside the ordinary expression.

AINSBURY and HAKIM

In the last office action the Examiner acknowledged that these two references do not "explicitly disclose an interface for allowing the user to specify in advance of a request where the application will display the requested information alongside and relative to the ordinary expression."

SHAFRON

The Examiner cites Shafron as "a method of dynamically controlling and displaying a toolbar menu in a graphical user interface." In particular, Shafron describes a controlling program that creates a library file including ActiveX controls or plug-in functionality. This reference describes creating a pull down menu 44 in the toolbar 22 of a browser. In any event, the Examiner acknowledges that Shafron (as well as the other cited art) does not disclose the "claim limitations of displaying the requested information alongside and relative to the ordinary expression."

CLAIM ANALYSIS

In the office action mailed June 8, 2005 the Examiner already conceded that all the cited references fail to show the "claim limitations of displaying the requested information alongside and relative to the ordinary expression," with the possible exception of Moseley.

After discussing Moseley in the interview of October 18, 2005 the Examiner identified two areas with potential for distinguishing over the cited art, including Moseley:

- (1) the capabilities to read requested data from a specific location on the desired information source (documents such as a spreadsheet, word processing, and web page); and
- (2) the ordinary expression is being read/written alongside the fetched requested data on the same window.

These two areas are emphasized in currently amended claim 1 which provides:

a query generator for (a) interposing in ordinary operations of the application in order to read the ordinary expression previously entered manually by the user in the array of data elements, and (b) requesting information from the data source based on the ordinary expression;

a composer for interposing in ordinary operations of the application in order to (a) receive requested information from the data source in response to said query generator and (b) transfer the requested information to the application for processing and display in the array of data elements alongside the ordinary expression before returning to a condition where the application operates autonomously;

Thus, with respect to the first area, Moseley does not suggest reading "the ordinary expression previously entered manually by the user in the array of data elements" and "requesting information from the data source based on the ordinary expression," as recited in amended claim 1. With respect to the second area, Moseley does not suggest operating to receive the "requested information from the data source [and] transfer the requested information to the application for processing and display in the array of data elements alongside the ordinary expression," as recited in amended claim 1.

The Examiner will notice how this reading and writing is performed at the same place; namely, among the "array of data elements." Claim 1 makes it clear that this array of data elements is in an ordinary workspace (as opposed to some space for setting up or designing forms). Specifically, the preamble of claim 1 sets up the definition of this "array of data elements" by reciting "an application that without enhancement is ordinarily capable of presenting information in an array of data elements."

The foregoing arguments, by themselves, are sufficient to distinguish over

Moseley without considering the nature of Moseley's "form design screen." In other words, Moseley's "form design screen" is irrelevant. However, amended claim 1 recites a unique interface that allows the user to specify where to read and write among the array of data elements. Specifically, amended claim 1 recites:

an interface for interposing in ordinary operations of the application in order to allow the user to specify in advance of a request (a) where in the array of data elements the query generator will find the ordinary expression, and (b) where in the array of data elements the requested information will be displayed relative to the ordinary expression.

Moseley's form design screen can set up writing operations, but cannot set up reading operations of the type disclosed in the present application. Specifically, the form design screen cannot be used to specify "where in the array of data elements [to] find the ordinary expression, and ... where in the array of data elements the requested information will be displayed relative to the ordinary expression," as recited in amended claim 1. Moseley simply has no teaching of this rather advanced feature.

The other independent claim, method claim 41, provides in pertinent part:

entering the ordinary expression in the array of data elements of the application in the ordinary unenhanced way for processing and display;

electronically reading the ordinary expression in the array of data elements by interposing in ordinary operations of the application;

requesting information from the data source based on the ordinary expression;

receiving requested information from the data source;

automatically transferring the requested information to the application for processing and display in the array of data elements alongside the ordinary expression before returning to a condition where the application operates autonomously;

As before, with respect to the first area, Moseley does not suggest "electronically

reading the ordinary expression in the array of data elements" and "requesting information from the data source based on the ordinary expression," as recited in amended claim 41. With respect to the second area, Moseley does not suggest "receiving requested information from the data source; [and] automatically transferring the requested information to the application for processing and display in the array of data elements alongside the ordinary expression," as recited in amended claim 41.

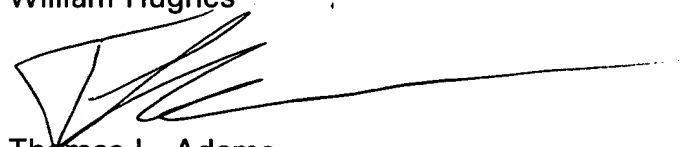
Therefore, amended claim 41 distinguishes over Moseley in a manner similar to amended claim 1. Also, amended method claim 41 further distinguishes over Moseley by concluding with a unique step that is similar to the "interface" of amended claim 1.

CONCLUSION

The other claims depend from these independent claims and distinguish over the cited art for at least the reasons given in connection with those independent claims.

It is believed that the foregoing fully responds to the objections and rejections entered by the Examiner and places this application in condition for allowance, which action is respectfully requested.

Respectfully submitted,
William Hughes

A handwritten signature in black ink, appearing to read 'Thomas L. Adams', with a long horizontal flourish extending to the right.

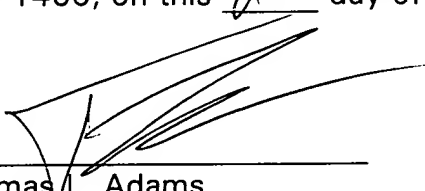
Thomas L. Adams
Registration No. 27,300
(973)463-0100



25

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid addressed to: Mail Stop No Fee Amendment; Commissioner for Patents; P.O. Box 1450; Alexandria, VA 22313-1450, on this 22nd day of February, 2006



Thomas L. Adams
Registration No. 27,300

CERTIFICATE OF TRANSMISSION UNDER 37 CFR 1.8

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office on this _____ day of February, 2006

Thomas L. Adams
Registration No. 27,300